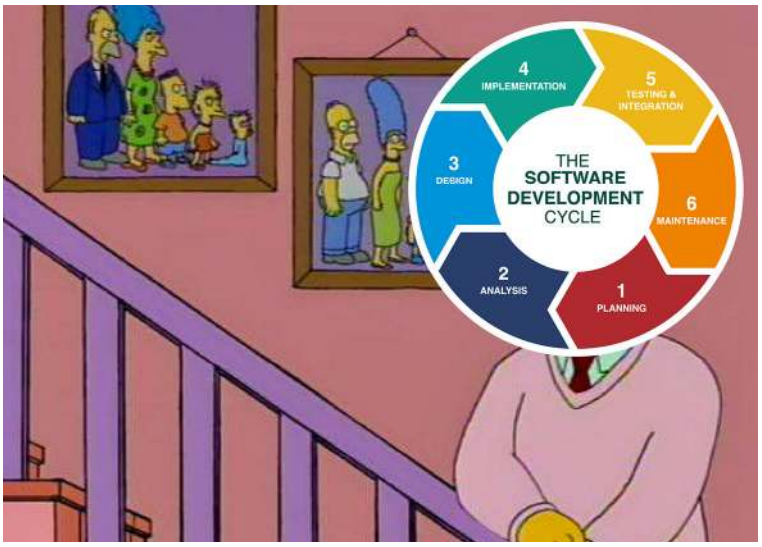
Sourcegraph

# How Developers think about Developer Productivity




*Hi, I'm the SDLC. You may remember me from such developer marketing campaigns as Shift Left, DORA the DevOps Explorer, and Agile Waterfalls.*

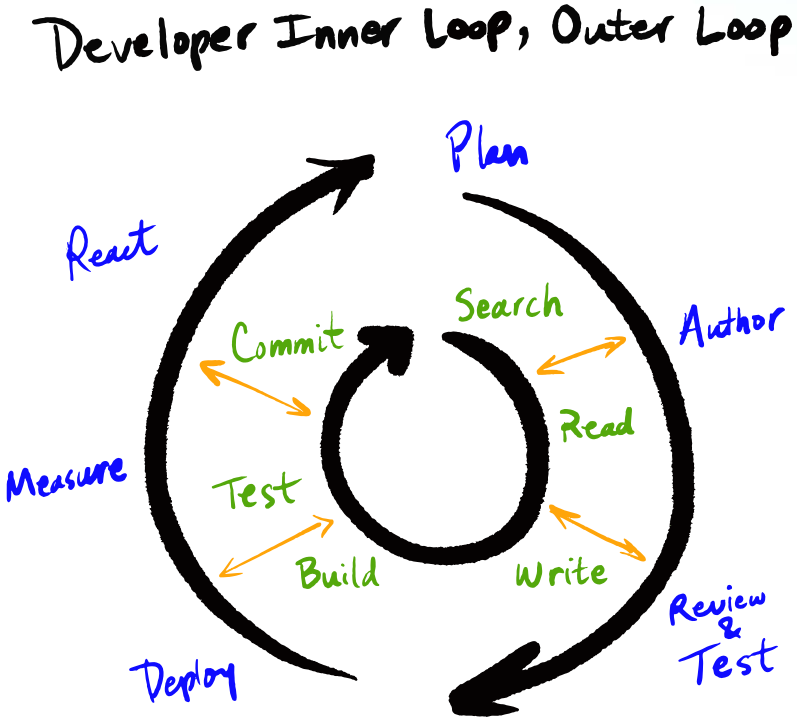
## The "inner loop"

The common picture of the software development process is the SDLC, which describes the many stages of software development, but it leaves out a crucial step: how the code is written.

**Picture a developer's work as two nested loops:**

- The outer loop roughly follows the SDLC, which happens at the level of sprints, projects, and releases.
- The inner loop is the read-write-run loop that happens many times a day when developers are busy understanding code, writing code, running tests, and repeating until they are happy.

Developers need to make organizations aware of this inner loop so that it can be understood and valued.



# Flow state

When working on code, developers can get into a kind of zen mode — where they become the work — and get SOOO much done. They sometimes call this…the flow state.

Flow state is the work process with minimum interruptions and maximum output. It speeds up the inner loop. To improve developers' productivity, organizations should optimize this state. Developers need time to think in order to work, the fewer interruptions the better.

Developers need time to think in order to work, the fewer interruptions the better.

Because the "inner loop" is such a common part of the development process, organizations need to talk about what they need to do to maximize its efficiency.

When organizations understand this, they can account for a developer's drop in productivity, like when they may need to wrap their head around a new library. Organizations can then do what they can to reduce unnecessary distractions.

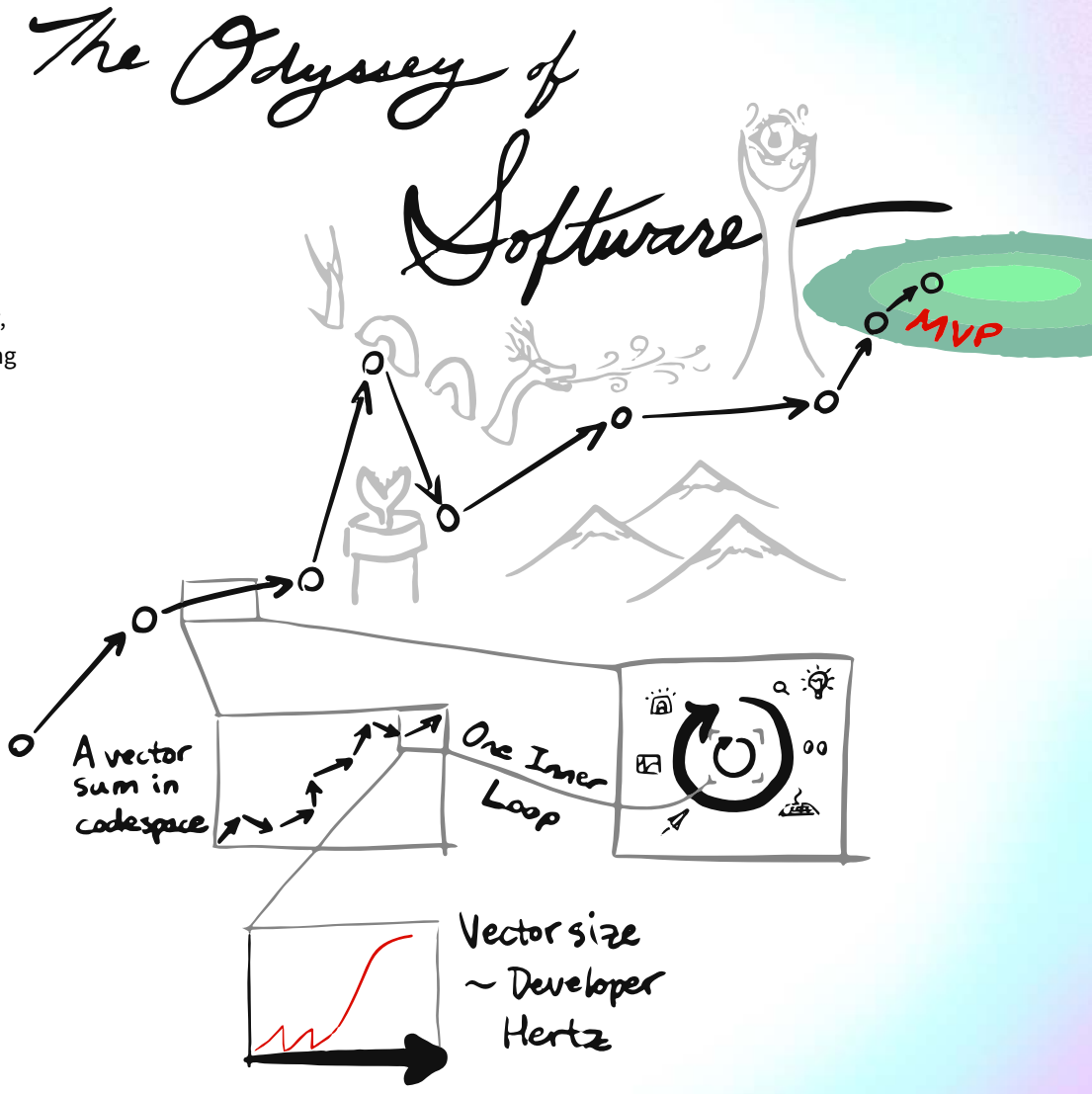
# Progress = flow state * technical direction

Once the developer's work process is understood, the next crucial step is to identify a direction to work towards.

Even if developers are working efficiently, their work will bear no fruit if they're going in the wrong technical direction.

The technical direction an organization takes – whether they use library X or Y, for example – can provide a critical shortcut to their MVP. Picking the wrong direction might mean they have to retrace their steps later.

Choosing a destination wisely helps developers know how to get there. The best choice is to break down a destination into a "zone of acceptable outcomes". Developers should get into the acceptable zone, and then improve on their position.

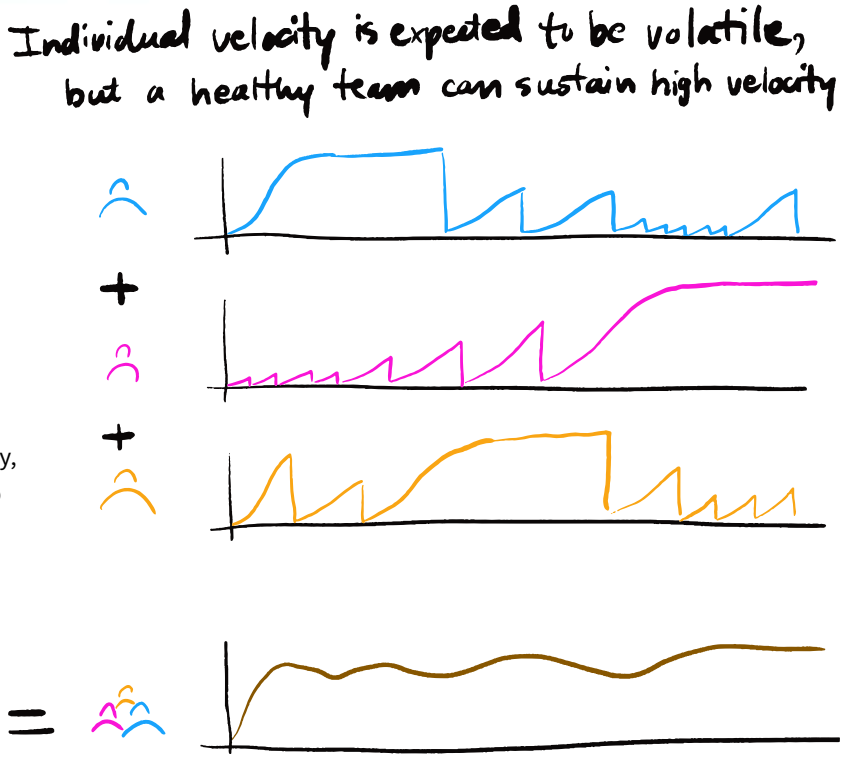


## Teamwork makes the dream work

An individual developer's progress tends to be choppy. There are inevitably days spent onboarding to unfamiliar code or when regularly scheduled programming is interrupted by external factors.

When developers work as a team, however, individual bumps smooth out when added together.

If a team suffers from a sustained period of low productivity, it's worthwhile to ask what factor caused a correlated drop in productivity across all members of the team.

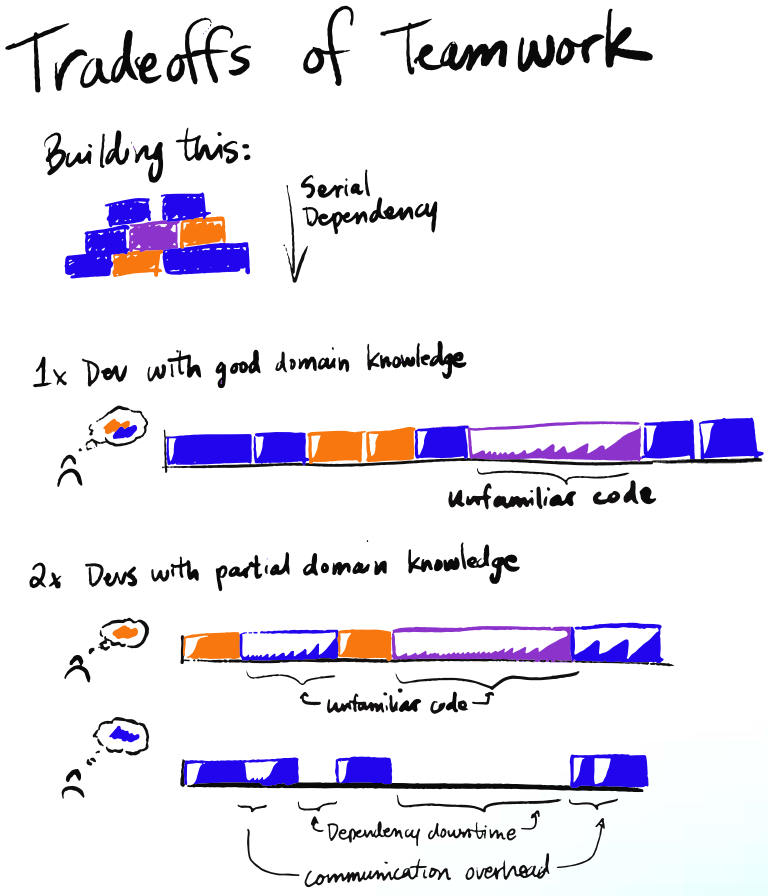



## Too many cooks

The more members there are on a development team, yes the more minds on the project, but other complex issues start to crop up too.

Communication overhead, work dependency structures, and the speed of context acquisition can all increase with the size of a team.

This is why most organizations choose to keep good developers rather than hiring more of them. This is also why they choose to invest in great tools to increase the productivity and coordination of their existing workers.



# Power to the people!

When developers' perspectives are left out of the picture, companies can fall into the trap of hiring too many people, creating more code and more problems.

To fix this, developers need to start to take control of the way their work is thought about and talked about.

This is just a start at bringing developer voices to the table. The more developers talk about their work, the more it will be understood and valued.

---

Sourcegraph's code intelligence platform is more than simply search. The platform drives velocity by helping development teams quickly get unblocked, save time resolving issues, and gain insights to make better decisions. Request a demo to learn more about ways we can help accelerate your development team.

Request a demo